REMARKS

Applicants respectfully request reconsideration of the present application based on the foregoing amendments and the following remarks. By this Amendment, claims 1, 58, 61 and 63 have been amended and claim 65 has been added. Upon entry of the amendment, claims 1-6 and 8-65 will remain pending in the application.

Allowable Subject Matter

It is believed that claims 2-6, 8-38, 40-47 and 50-64 are allowable because the Office Action did not specify any basis for the rejection of these claims (see MPEP § 707.07(i)). Applicants respectfully request Notice to that effect. Alternatively, any subsequent Office Action setting forth additional reasons why these claims are rejected should be non-final, because Applicants have not had an opportunity to respond to any basis for these rejections.

Claim Objections

Claim 1 stands objected to because the phrase "generating a description of the user-defined register file separate from and in addition to a description of the core register file in the description of the hardware implementation of the processor" is allegedly unclear.

Applicants respectfully submit that the language of claim 1 is clear to those skilled in the art, particularly in view of the specification. The language of claim 1 makes clear that a **user-defined** register file can be included in the description of the processor at the discretion of the user. In contrast, the predetermined **core** register file is always included in the description of the processor, albeit with a variable configuration. Specifically, the claim further requires a "configuration specification including a predetermined portion and a user-defined portion." The "predetermined portion" specifies a **configuration** of a predetermined core register file. The "user-defined portion" specifies **whether or not** to include a user-defined register file **in addition to** the core register file. (see, for example, the present specification on page 13, lines 3-5, discussing a TIE construct that allows a user to "designate a register file declared with regfile in addition to the core register file.") Accordingly, the claim makes clear that the hardware generation means is able to generate a description of a processor **with or without** a user-defined register file, but always with a predetermined core register file. (see, for example, the present specification at page 13, line 35 - page 14, line 2 of the specification: "[a] 16-entry, 8-bit [user-defined] register file is created ... and two instructions are defined that operate on these registers.") To make this feature of the invention even more clear, the claim requires that the hardware generation means is able to generate "a description of the user-defined register file **separate from and in addition to** a description of the core register file."

In one example, of the invention, the description of the hardware implementation of the processor is in a register transfer level (RTL) hardware description language (see claim 3). Those skilled in the art understand that such a hardware description can include several portions describing different portions of a hardware circuit. As for claim 1, those skilled in the art would understand that claim 1 requires generating a hardware description that may include different and **separate** portions for a **core** register file and a **user-defined** register file. Accordingly, it is believed that the explicit language of claim 1 would be clear to those skilled in the art.

Claim 49 also stands objected to because the phrase "comparing results of simulations" is allegedly unclear. More completely, claim 49 requires:

> a cosimulation means for operating the hardware simulation means and the software simulation means and comparing results of simulations therefrom to establish correspondence between the hardware description of the extensible processor and the software reference model of the extensible processor.

It is believed that those skilled in the art would readily understand that "results of simulations therefrom" are **results of operating the hardware simulation means and the software simulation means**, which the cosimulation means **compares to establish a correspondence between the hardware description and the software reference model** of the extensible processor. An example of the cosimulation techniques of the present invention is set forth in the present specification at, for example, page 106-111. Accordingly, it is believed the language of claim 49 would be clear to those skilled in the art, particularly in view of the descriptions set forth in the specification.

Because the language of the claims are considered clear in view of the specification to one skilled in the art, it is respectfully submitted that no amendments to claims 1 and 49 are required, and the objection to the claims should be withdrawn.

Claim Rejections Under 35 U.S.C. § 102(e)

Claims 1-6 and 8-64 stand rejected under 35 U.S.C. 102(e) as being unpatentable over U.S. Patent No. 6,385,757 to Gupta et al. ("Gupta"). For reasons set forth below, this rejection is respectfully traversed.

<u>Amended Independent Claim 1 Patentably Defines Over Gupta</u>

Amended independent claim 1 requires, *inter alia*,

> hardware generation means for, based on a configuration specification including a predetermined portion and a user-defined portion, generating a description of a hardware implementation of the processor, the predetermined portion specifying a configuration of a core register file, and the user-defined portion specifying **<u>whether to include a user-defined register file supporting a freely user-defined data type</u>** in the processor in addition to the core register file;
>
> ...
>
> wherein the hardware generation means includes register generation means for, based on the user-defined portion of the configuration specification, generating a description of the user-defined register file separate from and in addition to a description of the core register file in the description of the hardware implementation of the processor; and

The clear and explicit language of amended claim 1 thus requires that the **<u>user-defined portion</u>** of the configuration specification includes the ability to specify **<u>whether</u>** a user-defined register file **<u>supporting a freely user-defined data type</u>** should be included in the processor **<u>in addition to</u>** the core register file (see the present specification at, for example, page 12, line 8 to page 14 line 11, describing TIE specification of a new operand, and creating a register file to support it).

At best, Gupta only discloses a system that allows design of a processor having register files with certain predetermined data types such as integer, floating point, predication, etc. It does not include the ability to specify a user-defined register file supporting a freely user-defined data type, and the ability to generate a description of a processor having a core register file configured by a user as well as the freely user-defined register file separate from and in addition to a description of the core register file.

For example, Gupta teaches at col. 7, line 55 – col. 8, line 5 that:

> In our implementation, the register file specification includes the following:
>
> 1. Register file types--e.g., integer, floating-point, predicate, branch, etc.
>
> 2. The number of register files of each type.
>
> 3. The number of registers in each file. Registers are divided into static and rotating. Thus, it specifies the number of static registers and number of rotating registers.
>
> 4. The bit-width of registers in a file.
>
> 5. Presence or absence of speculative tag bit.
>
> a specification of the desired ILP constraints, making use of some form of concurrency sets, exclusion sets or a combination of concurrency and exclusion sets, that specifies which sets of operation groups/opcodes can be issued concurrently; and
>
> other optional architecture parameters, e.g., presence/absence of predication, speculation, etc.

This register file specification allows only adjustment of a predetermined set of register files having predetermined (e.g. integer, floating point, etc.) data types. There is nothing to suggest specifying **a freely user-defined data type** in addition to the predetermined register files provided for in the implementation.

For at least these additional reasons, amended independent claim 1, together with rejected claims 2-6 and 8-38 that depend therefrom, patentably defines over the prior art and the § 102 rejection of the claims should be withdrawn.

Independent Claim 39 Patentably Defines Over Gupta

Independent claim 39 requires, *inter alia*,

> hardware generation means for, **based on a configuration specification** including a predetermined portion and a user-defined portion, generating a description of a hardware implementation of the processor;
>
> . . .
>
> wherein the configuration specification includes a **statement specifying scheduling information of instructions used in the software development tools;**
>
> the hardware generation means is for, **based on the statement in the configuration specification, determining whether and how to generate a description** of at least one of pipeline logic, pipeline stalling logic and instruction rescheduling logic

The language of claim 39 thus explicitly requires that: (1) the **configuration specification** includes the ability to specify **scheduling information of instructions**; and (2) the hardware generation means must be able to determine **whether and how** to generate a description of instruction scheduling logic in the description of the hardware implementation of the processor based on the instruction scheduling information in the configuration specification.

Gupta discloses nothing about generating a hardware description of a processor, in which certain types of instruction scheduling logic may be optionally included in the processor hardware description based on scheduling information in a configuration specification.

The Office Action relies on col. 4, lines 50-55 of Gupta as allegedly describing the claimed configuration specification and hardware generation means:

> In some design scenarios, the system uses the extracted MDES to re-target a compiler. The re-targeted compiler schedules an application program (or set of application programs) based on the MDES and provides operation issues statistics. The system uses these statistics to select custom instruction templates that optimize the instruction format design for the application program or set of programs.

The first passage merely describes Gupta's MDES extractor, which programmatically extracts a machine description of a processor from various sources such as the abstract ISA specification of the processor. Gupta's system further allows for re-targeting a compiler for the target processor based on information in the MDES, and the compiler allows for an application to be scheduled for the target processor in appropriate machine instructions. Statistics can then be generated to optimize application program instructions for the target processor.

These teachings are almost directly opposite from the requirements of claim 39, which requires a **hardware generation means** to optionally include **logic** in the hardware description of a processor **based on an instruction scheduling information statement in a configuration specification** of a processor. This passage of Gupta merely describes re-targeting a compiler for scheduling applications on a target processor designed by an automated processor design system. Applicants respectfully submit that this passage says nothing about a configuration specification including an **instruction scheduling information statement** nor **optionally generating a description of logic based on** such a statement at all, much less the explicit types of logic required by claim 39.

The Office Action also relies on col. 26, lines 40-45 as allegedly describing the claimed configuration specification and hardware generation means:

> The time of use is determined by taking into account the latency of the operation that is provided in the operation's mini-MDES and any pipeline latches encountered during the structural traversal. In this manner the composite reservation pattern of each alternative of the operation is built by combining information from the mini-MDES and the structural connectivity of the target machine.

This passage merely describes how various functional units, each with their own "mini-MDES" is added to the extracted MDES of a target processor. Applicants respectfully submit that this passage says nothing about **an instruction scheduling information statement** in a configuration specification, much less **determining whether and how to generate** a hardware description based on such a statement at all, much less as required in claim 39.

For at least these additional reasons, independent claim 39, together with rejected claims 40-48 and 58-60 that depend therefrom, patentably defines over the prior art and the § 102 rejection of the claims should be withdrawn.

Independent Claim 45 Patentably Defines Over Gupta

Independent claim 45 requires, *inter alia*,

> **document generation means for generating documentation of a processor instruction set** described by the configuration specification **based on the configuration specification.**

The Office Action correctly fails to point to any description in Gupta corresponding to this subject matter.

For at least these additional reasons, independent claim 45, together with rejected claims 46-47 and 61-62 that depend therefrom, patentably defines over the prior art and the § 102 rejection of the claims should be withdrawn.

Independent Claim 48 Patentably Defines Over Gupta

Independent claim 48 requires, *inter alia*,

> hardware generation means for, **based on a configuration specification** including a predetermined portion and a user-defined portion, **generating a description of a hardware implementation of the processor**; and
>
> . . .
>
> wherein the user-defined portion of **the configuration specification includes a user-defined specification of a processor exception and when a processor instruction raises the exception**; and
>
> the hardware generation means **includes user-defined exception support generating means for generating hardware supporting that user-defined exception** as part of the processor hardware implementation.

The language of claim 48 thus explicitly requires that: (1) the **configuration specification** includes the ability to optionally include **user-defined processor instruction exceptions**; and (2) the hardware generation means must be able to **generate hardware supporting the user-defined exception** in the description of the hardware implementation of the processor based on the configuration specification.

The Office Action relies on col. 10, lines 17-29 of Gupta which discloses:

> The instruction sequencer is the control logic that interfaces with the control logic of the IUdatapath and specifies the sequence of instructions to be fetched from the instruction cache. It manages a memory address register (MAR) that holds the memory address of the next instruction to be fetched from.the instruction cache, and the Program Counter, identifying the next instruction to be executed in the processor. The control ports of the sequencer interface with the control ports of the IUdatapath control logic. The sequencer is also responsible for interfacing with the branch functional unit and for managing events such as interrupts and exceptions. The sequencer is a generic macrocell.

This passage merely describes control logic in Gupta's processor that is responsible for managing events such as interrupts and exceptions. Applicants respectfully submit that this passage says nothing about a configuration specification or a hardware generation means at all, much less (1) a **configuration specification** which includes the ability to optionally include **user-defined processor instruction exceptions**; and (2) a hardware generation means that is able to **generate hardware supporting the user-defined exception** in the description of the hardware implementation of the processor based on the configuration specification, as required by claim 48.

For at least these additional reasons, independent claim 48, together with rejected claims 63-64 that depend therefrom, patentably defines over the prior art and the § 102 rejection of the claims should be withdrawn.

Independent Claim 49 Patentably Defines Over Gupta

Independent claim 49 requires, *inter alia*,

> hardware simulation means for executing a hardware description of an extensible processor;
> software simulation means for executing a software reference model of the extensible processor; and
> **cosimulation means for operating the hardware simulation means and the software simulation means** and comparing results of simulations therefrom to **establish correspondence between the hardware description of the extensible processor and the software reference model of the extensible processor**.

Gupta discloses nothing about a cosimulation means that operates a hardware simulation means and software simulation means, much less one that establishes correspondence between the hardware description and the software reference model as required by claim 49.

The Office Action relies on this passage of Gupta (col. 46, lines 25-30) as allegedly describing the claimed cosimulation means:

> After the concurrency matrix is generated, the system compares the rows in the concurrency matrix to identify equivalent operation groups. The super groups are formed from the equivalent operation groups. Two operation groups are said to be equivalent if they have the same set of concurrency neighbors. Note that two mutually exclusive operation groups that have the same set of concurrency neighbors can replace each other in any template without violating any exclusion constraint and therefore can be treated equivalently. Similarly, two concurrent operation groups that have the same set of concurrency neighbors (other than themselves) can always be placed together in a template without violating any exclusion constraints and therefore can be treated equivalently.

This passage merely describes part of Gupta's process for producing instruction templates that support all sets of operation groups that can be issued concurrently (see col. 44, lines 53-66).

Applicants respectfully submit that this passage says nothing about a hardware and software simulation means at all, much less a cosimulation means as required in claim 49.

For at least these additional reasons, independent claim 49, together with rejected claims 50-57 that depend therefrom, patentably defines over the prior art and the § 102 rejection of the claims should be withdrawn.

Dependent Claims 2-6, 8-38, 40-44, 46-47 and 50-64 Patentably Define Over Gupta

Claims dependent on independent claims 1, 39, 45, 48 and 49 are patentable at least for the reasons set forth above. The Office Action correctly fails to point to any teaching in Gupta that meets the limitations of any dependent claim.

For example, claim 2 requires that software related to the user-defined processor register file includes an instruction for accessing elements in the register file according to a field of the instruction. The Office Action correctly fails to point to any teaching in Gupta that meets these additional limitations.

Claim 3 requires that the hardware generation means is for generating at least part of the description of the hardware implementation in a register transfer level hardware description language. The Office Action correctly fails to point to any teaching in Gupta that meets these additional limitations.

Claim 4 requires that the configuration specification defines the user-defined register file using a statement specifying the width of elements in the user-defined register file. The Office Action correctly fails to point to any teaching in Gupta that meets these additional limitations.

Claim 5 requires that the configuration specification defines the user-defined register file using a statement specifying the number of elements in the user-defined register file. The Office Action correctly fails to point to any teaching in Gupta that meets these additional limitations.

Claim 6 requires that the hardware generation means is for determining a number of at least one of read ports and write ports of the user-defined register file independently of the configuration specification. The Office Action correctly fails to point to any teaching in Gupta that meets these additional limitations.

Claim 8 requires that the hardware generation means is for generating, as part of the processor hardware implementation description, a description of logic to assign write ports of the user-defined register file to instruction operands to minimize data staging costs. The Office Action correctly fails to point to any teaching in Gupta that meets these additional limitations.

Claim 9 requires that the hardware generation means is for generating pipeline logic for accessing the user-defined register file. The Office Action correctly fails to point to any teaching in Gupta that meets these additional limitations.

Claim 10 requires that read ports for the user-defined register file are read in the earliest stage of any instruction that uses them as a source operand. The Office Action correctly fails to point to any teaching in Gupta that meets these additional limitations.

Claim 11 requires that write ports for the user-defined register file are read in the latest stage of any instruction that uses it as a destination operand or in an instruction commit stage if later. The Office Action correctly fails to point to any teaching in Gupta that meets these additional limitations.

Claim 12 requires that the hardware generation means is for generating, as part of the hardware implementation of the processor, logic to provide a read port for the user-defined register file for each field, within an instruction accessing the user-defined register file, used to select a source operand from the user-defined register file. The Office Action correctly fails to point to any teaching in Gupta that meets these additional limitations.

Claim 13 requires that the hardware generation means is for generating, as part of the hardware implementation of the processor, bypass logic for accessing the user-defined register file. The Office Action correctly fails to point to any teaching in Gupta that meets these additional limitations.

Claim 14 requires that the hardware generation means is for generating the interlock logic for a given pipeline of the processor described by the configuration specification based on instruction operand and state usage descriptions in the configuration specification. The Office Action correctly fails to point to any teaching in Gupta that meets these additional limitations.

Claim 15 requires that the hardware generation means is for generating, as part of the hardware implementation of the processor, interlock logic for accessing the register file. The Office Action correctly fails to point to any teaching in Gupta that meets these additional limitations.

Claim 16 requires that the hardware generation means is for generating the interlock logic based on scheduling information in the configuration specification. The Office Action correctly fails to point to any teaching in Gupta that meets these additional limitations.

Claim 17 requires that the hardware generation means is for generating the interlock logic for a given pipeline of the processor described by the configuration specification based on instruction operand and state usage descriptions in the configuration specification. The Office Action correctly fails to point to any teaching in Gupta that meets these additional limitations.

Claim 18 requires that the hardware generation means is for generating the processor hardware implementation description to use at least one portion of processor logic described by the predetermined portion of the configuration specification to support access of the user-defined register file. The Office Action correctly fails to point to any teaching in Gupta that meets these additional limitations.

Claim 19 requires that the at least one portion of processor logic includes address computation logic. The Office Action correctly fails to point to any teaching in Gupta that meets these additional limitations.

Claim 20 requires that the address computation logic includes address adder logic. The Office Action correctly fails to point to any teaching in Gupta that meets these additional limitations.

Claim 21 requires that the at least one portion of processor logic includes data alignment logic shared between the portions of the processor corresponding to predetermined and user-defined portions of the configuration specification. The Office Action correctly fails to point to any teaching in Gupta that meets these additional limitations.

Claim 22 requires that the at least one portion of processor logic is a data memory. The Office Action correctly fails to point to any teaching in Gupta that meets these additional limitations.

Claim 23 requires that the user-defined portion of the configuration specification includes a description of an instruction which conditionally writes to the user-defined register file. The Office Action correctly fails to point to any teaching in Gupta that meets these additional limitations.

Claim 24 requires that the software generation means is for generating, as part of the software relating to the user-defined register file, diagnostic tests for design verification and manufacturing of the processor based on the configuration specification. The Office Action correctly fails to point to any teaching in Gupta that meets these additional limitations.

Claim 25 requires that the configuration specification includes both reference and implementation semantics for instructions of the processor; and the reference semantics can be used to verify design correctness of the implementation semantics. The Office Action correctly fails to point to any teaching in Gupta that meets these additional limitations.

Claim 26 requires that the processor instruction set description language includes instruction test cases; and the software generation means is for generating diagnostics for the test cases. The Office Action correctly fails to point to any teaching in Gupta that meets these additional limitations.

Claim 27 requires that the software generation means is for automatically generating test vectors by sampling operands to instructions in the processor instruction set description language while running an application. The Office Action correctly fails to point to any teaching in Gupta that meets these additional limitations.

Claim 28 requires that the software generation means is for generating at least a portion of an operating system as part of the software relating to user-defined states and register files. The Office Action correctly fails to point to any teaching in Gupta that meets these additional limitations.

Claim 29 requires that the generated portion of the operating system includes save and restore sequences for processor state. The Office Action correctly fails to point to any teaching in Gupta that meets these additional limitations.

Claim 30 requires that the save and restore sequences are generated with respect to interdependencies of component states and is valid for those interdependencies. The Office Action correctly fails to point to any teaching in Gupta that meets these additional limitations.

Claim 31 requires that the operating system is capable of saving less than an entirety of processor state in accordance with a dynamic reference by a new task after switching contexts. The Office Action correctly fails to point to any teaching in Gupta that meets these additional limitations.

Claim 32 requires that the user-defined portion of the configuration specification defines a software data type not found in the predetermined portion of the configuration specification; and the compiler supports the software data type. The Office Action correctly fails to point to any teaching in Gupta that meets these additional limitations.

Claim 33 requires that the software generation means is for generating at least one of a compiler, a linker, a simulator and a debugger as part of the software relating to the user-defined register file. The Office Action correctly fails to point to any teaching in Gupta that meets these additional limitations.

Claim 34 requires that the software generation means is for generating a compiler as part of the software relating to the user-defined register file; and the compiler is capable of allocating program variables to registers in the user-defined register file. The Office Action correctly fails to point to any teaching in Gupta that meets these additional limitations.

Claim 35 requires that the compiler is further capable of loading a value from memory into a register in the user-defined register file, and storing a value in a register of the user-defined register file into memory. The Office Action correctly fails to point to any teaching in Gupta that meets these additional limitations.

Claim 36 requires that the compiler is further capable of moving a value from one register in a user-defined register file to another register in a user-defined register file. The Office Action correctly fails to point to any teaching in Gupta that meets these additional limitations.

Claim 37 requires that the compiler is for using scheduling information in the configuration specification to determine stall cycles of instructions in the software generated by the software generation means which access the user-defined register file. The Office Action correctly fails to point to any teaching in Gupta that meets these additional limitations.

Claim 38 requires that the software generation means is for automatically generating a monitor to check for coverage of bypass paths. The Office Action correctly fails to point to any teaching in Gupta that meets these additional limitations.

Claim 40 requires that the scheduling information includes a statement that an operand of an instruction enters a pipeline of the processor at a given stage. The Office Action correctly fails to point to any teaching in Gupta that meets these additional limitations.

Claim 41 requires that the scheduling information includes a statement that an operation of an instruction exits a pipeline of the processor at a given stage. The Office Action correctly fails to point to any teaching in Gupta that meets these additional limitations.

Claim 42 requires that the software generated by the software generation means includes a compiler which uses instructions described in the user-defined portion of the configuration specification; and the compiler uses the scheduling information during instruction scheduling to schedule the instructions described in the user-defined portion of the configuration specification. The Office Action correctly fails to point to any teaching in Gupta that meets these additional limitations.

Claim 43 requires that the configuration specification includes a description of an instruction which requires a plurality of processor cycles to be processed. The Office Action correctly fails to point to any teaching in Gupta that meets these additional limitations.

Claim 44 requires that the configuration specification includes a description of an instruction's semantics which is independent of a target pipeline of the processor; and the hardware generation means is for generating as part of the processor hardware implementation a pipeline based on a pipeline description separate from the instruction semantics. The Office Action correctly fails to point to any teaching in Gupta that meets these additional limitations.

Claim 46 requires that the document generation means is for using reference semantics of instructions defined in the configuration specification to generate the processor instruction set documentation. The Office Action correctly fails to point to any teaching in Gupta that meets these additional limitations.

Claim 47 requires that the user-defined portion of the configuration specification contains reference semantics of an instruction defined therein and a user-defined specification of at least one of a synopsis and a text description for the user-defined instruction; and the document generation means is for using the at least one of the synopsis and the text description to generate documentation of the processor instruction set. The Office Action correctly fails to point to any teaching in Gupta that meets these additional limitations.

Claim 50 requires that the hardware description of the extensible processor includes a description of hardware supporting one or more user-defined instructions. The Office Action correctly fails to point to any teaching in Gupta that meets these additional limitations.

Claim 51 requires that the software reference model includes a model of the extensible processor including the one or more user-defined instructions. The Office Action correctly fails to point to any teaching in Gupta that meets these additional limitations.

Claim 52 requires that the hardware description of the extensible processor executed by the hardware simulation means includes a description of one or more user-defined states. The Office Action correctly fails to point to any teaching in Gupta that meets these additional limitations.

Claim 53 requires that the software reference model executed by the software simulation means includes a model of the extensible processor including the one or more user-defined states. The Office Action correctly fails to point to any teaching in Gupta that meets these additional limitations.

Claim 54 requires that the cosimulation means establishes correspondence of all the processor states, including the user-defined states, between the hardware description of the extensible processor and the software reference model of the extensible processor. The Office Action correctly fails to point to any teaching in Gupta that meets these additional limitations.

Claim 55 requires that the hardware description of the extensible processor executed by the hardware simulation means includes a description of one or more user-defined register files. The Office Action correctly fails to point to any teaching in Gupta that meets these additional limitations.

Claim 56 requires that the software reference model executed by the software simulation means includes a model of the extensible processor including the one or more user-defined register files. The Office Action correctly fails to point to any teaching in Gupta that meets these additional limitations.

Claim 57 requires that the cosimulation means establishes correspondence of all the processor register files, including the user-defined register files, between the hardware description of the extensible processor and the software reference model of the extensible processor. The Office Action correctly fails to point to any teaching in Gupta that meets these additional limitations.

Claim 58 requires that the user-defined portion of the configuration specification supports one or more user-defined instructions that are added to a pre-defined instruction set of the processor. The Office Action correctly fails to point to any teaching in Gupta that meets these additional limitations.

Claim 59 requires that the instructions used in the software development tools include one or more user-defined instructions. The Office Action correctly fails to point to any teaching in Gupta that meets these additional limitations.

Claim 60 requires that the pipeline logic, pipeline stalling logic and instruction rescheduling logic are for supporting one or more user-defined instructions. The Office Action correctly fails to point to any teaching in Gupta that meets these additional limitations.

Claim 61 requires that the user-defined portion of the configuration specification supports one or more user-defined instructions that are added to a pre-defined instruction set of the processor. The Office Action correctly fails to point to any teaching in Gupta that meets these additional limitations.

Claim 62 requires that the processor instruction set includes a predetermined portion and a user-defined portion. The Office Action correctly fails to point to any teaching in Gupta that meets these additional limitations.

Claim 63 requires that the user-defined portion of the configuration specification supports one or more user-defined instructions that are added to a pre-defined instruction set of the processor. The Office Action correctly fails to point to any teaching in Gupta that meets these additional limitations.

Claim 64 requires that the processor instruction is a user-defined instruction. The Office Action correctly fails to point to any teaching in Gupta that meets these additional limitations.

Accordingly, for at least these additional reasons the § 102 rejection of the claims should be withdrawn.

Conclusion

All objections and rejections having been addressed, it is believed the present application is in condition for allowance, and Notice thereof is earnestly solicited. If any issues remain which the Examiner feels may be resolved through a telephone interview, s/he is kindly requested to contact the undersigned at the telephone number listed below.

Respectfully submitted,
PILLSBURY WINTHROP SHAW PITTMAN LLP

Date: April 13, 2005

Mark J. Danielson — 40,580 Reg. No.
Telephone: (650) 233-4777
Facsimile: (650) 233-4545
Reply to Customer No. 27,498